



Generando getter setter

Transcripción

[00:00] Entonces ¿será que de verdad necesito escribir `getAgencia`, `setAgencia` para cada uno de mis métodos de esta clase? Hay un atajo aquí también en Eclipse que nos va a ayudar mucho con ese código, y eso lo vamos a ver aquí en el caso de nuestro objeto cliente. ¿Qué sucede con nuestro cliente?

[00:28] Ya sabemos que para asignarle valor a nuestros campos lo hacemos a través de métodos `get` y métodos `set`. En el caso de nuestro cliente, sí vamos a tener que asignarle un cliente, por lo tanto repitiendo el ejercicio anterior vamos a comenzar a escribir `setTitu`, "`Ctrl + espacio`" y él nuevamente me sugiere: ¿será que tú quieres un `set titular`?

[00:55] Le digo que sí, quiero un `set titular`, y él ya me autogenera todo el código para el `set`. Vamos aquí y vamos a crear un nuevo cliente. Vamos a cerrar esta clase aquí y vamos a crear otra nueva. Le vamos a dar un `new class` y esa clase se va a llamar `PruebaEncapsulamiento`. Le damos `finish`, nuevamente nuestro ya conocidísimo método `main`, creamos nuestra cuenta obviamente.

[01:31] Cuenta, "`Ctrl + espacio`", cuenta, igual `new` "`Ctrl + espacio`", cuenta. Listo. ¿Se dan cuenta ya cómo podemos escribir todo este código en cuestión de segundos? Es la ventaja de usar un IDE tan bueno como Eclipse. Ahora, en el caso del cliente, yo ya voy a decirle cliente, igual `new cliente`. Punto y coma. Y en el cliente aún yo puedo decirle que `cliente.nombre` sea igual a Diego.

[02:22] Puedo decirle que `cliente.documento` sea igual digamos, no sé, que acepte caracteres alfanuméricos, que acepte cualquier cosa, y tenemos el

mismo problema en el caso de la cuenta. Básicamente nuestros atributos están al descubierto, están desprotegidos. ¿Cómo solucionamos el problema? Utilizando los modificadores de acceso que están dentro de cliente y ahora lo nombramos todo privado.

[02:43] Private. Copiamos aquí también, private. Ahora todos nuestros atributos de cliente están protegidos. ¿Qué va a pasar acá? No va a compilar el código porque nos va a decir "Yo no tengo acceso a esta variable". Recuerden que los métodos y campos privados son accesibles únicamente dentro de la clase que los está declarando.

[03:16] Entonces, ¿qué necesitamos? Generar nuestros getters y setters. ¿Será que necesito escribir getNombre y así sucesivamente para cada valor? Eclipse sabe que este es un caso muy repetitivo y también nos ayuda. Vamos aquí a la pestaña source y aquí abajo tiene Generate Getters and Setters. Y eso es lo que necesitamos, queremos hacerle método get y set para cada variable.

[03:48] Le damos que sí. Marcamos las variables que deseamos para hacerle Getter y Setter. Por ejemplo yo quiero que todos tengan y en este caso marco todos. ¿Qué pasaría si, yendo a nuestro caso anterior, en la cuenta no deseamos pues que teléfono tenga un set. Pues lo podemos desmarcar aquí, por lo tanto él solamente va a generar un método get.

[04:15] Por lo general, cuando autogeneras código, casi siempre marcas todo y generas el código get, set, para cada variable, aún cuando no lo necesitas. Entonces un consejo que sí les doy es: "miren bien si de verdad necesitan ese método set, ese método get, se van a dar cuenta con el paso del tiempo que no todas las variables privadas necesitan ser accesibles".

[04:40] Le damos generar y vemos aquí cómo él ya nos generó todo este código que está aquí abajo. Perfecto. Volvemos aquí, encapsulamiento, y aquí le damos un setNombre, como ya aprendimos, y le ponemos nuestro parámetro

adentro. Y ya el código ya va compilando de igual manera aquí. SetDocumento y el ya va a aceptar ese número de documento aquí adentro.

[05:19] Y así sucesivamente podemos ir modificando. Incluso si nosotros tuviéramos una expresión regular para validar el formato del documento, alguna cosa así, el set es un claro candidato a hacer esa validación para el documento.